AIAA 97-0333

An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids

R.F. Tomaro and W.Z. Strang
Interdisciplinary and Applied CFD Group
Wright Laboratory
Wright-Patterson AFB, OH

L.N. Sankar
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids

Robert F. Tomaro and William Z. Strang*
Interdisciplinary and Applied CFD Group
Wright Laboratory
Wright-Patterson AFB, OH 45433-7913

Lakshmi N. Sankar
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0150

Abstract

An implicit unstructured grid algorithm for solving 2-D and 3-D unsteady compressible viscous flows is presented. It is an extension of the COBALT finite volume unstructured grid code, and can handle a combination of cells of arbitrary type. The explicit first to fourth order accurate Runge-Kutta time integration algorithm in the original solver is replaced with the solution of an implicit system of nonlinear algebraic equations for the flow variables. This nonlinear system is first linearized about the flow state at an earlier time level (or iteration), and the resulting system is inverted using a two pass scheme. During the first pass, the flow property changes at the cell centers are updated first by ascending a cell reorder array; during the second pass, the cells are updated by descending the cell reorder array. In unsteady applications, a Newton iteration scheme is used at every time step to eliminate the errors associated with the linearization and the errors associated with the two pass scheme. Results for internal and external; steady and unsteady; 2-D, 3-D and axi-symmetric flows are given.

Introduction

One of the challenging problems currently facing the fighter aircraft industry is the ability to solve the 3-D flow equations over complete aircraft configurations. Such simulations can assist the designer in component design, in quantifying vortex lift enhancement, and in assessing the effects of vortex flow over the wing on the vertical tail airloads. Simulations involving unsteady airloads are important in determining structural strength through aeroelastic analysis. Unstructured grid based methods have been the preferred tools for modeling combat aircraft, because of the speed with which grids can be generated around these complex configuration compared to structured grids. Full aircraft simulations using unstructured grids have been successfully done[1].

Most finite volume unstructured algorithms are explicit. These methods are ideal for design, but are not suitable for time-dependent flows[2-4], or aeroelastic analysis because of their stiffness. Implicit methods are needed for specific time steps and/or very large time steps to make solving time dependent flows tractable.

Although many of the unstructured grid schemes have only been applied to steady problems[5-8], relatively few time-accurate simulations have been carried out[9-10]. Most of these implicit schemes use Roe's approximate Riemann solver or the Van Leer flux splitting scheme, and a diagonalization of the Euler Jacobians as a basis for the time integrations. Batina has used a point Jacobi, Gauss-Seidel, and a point implicit procedure to integrate in time. Venkatakrishnan and Mavriplis[11] have used GMRES with various preconditioning matrices as iterative techniques to integrate the equations in time.

The present study is aimed at the development of an unsteady unstructured solver that can be used for the entire flow regime of interest to the aircraft industry, from low subsonic speeds to hypersonic speeds. The flow solver uses the approximate Riemann solver of Collela[12] which is stiffer than the Roe solver, but is believed to be more accurate in the high speed regime. It is well known that accurate capturing of the viscous shear layer requires the use of hexahedral cells in the boundary layer. Therefore, the flow solver permits a combination of hexahedral (prismatic) cells and tetrahedra. Finally, the one-equation turbulence model used in this solver had to be integrated using an implicit scheme. The implicit solver had to be able to handle these new features, and attendant complexities.

---

*Aerospace Engineers
Regents' Professor, Senior Member AIAA

Mathematical and Numerical Formulation

Original COBALT Code

The original unstructured code COBALT[13] solves the 3-D unsteady Navier-Stokes equations which are first cast into integral form using the divergence theorem. The original solver is based on the first-order accurate in time and space, exact Riemann solver due to Godunov[14]. Exact Riemann solvers are very expensive, so the approximate Riemann solution method due to Collela in combination with the iterative method of Gottlieb and Groth[15] is utilized. Second order accuracy in space is achieved by assuming that the flow properties vary trilinearly within each cell. The flow gradients within a cell are computed by a least squares method. Second, third and fourth order accuracy in time is achieved using a multi-stage Runge-Kutta integration.

COBALT can solve 2-D, axi-symmetric, and 3-D Euler and Navier-Stokes problems. As stated earlier, the grid may be composed of any number of cells of arbitrary types. Different cell types are permitted with the same grid. In addition, the grid can be decomposed into sub-domains, called zones, to permit the solution of problem that are too large to reside in main memory.

Implicit Modifications

The present scheme is similar to two pass Gauss-Seidel schemes currently employed in some structured grid codes. There are several significant variations necessitated by the arbitrary numbering of cells in unstructured grids.

The Navier-Stokes equations in integral form are given by:

$$\frac{\partial}{\partial t}\iiint_{Vol}\mathbf{q}\,dVol+\oiint_{S}\left(\mathbf{F}\hat{i}+\mathbf{G}\hat{j}+\mathbf{H}\hat{k}\right)\cdot\mathbf{dS}=\oiint_{S}\left(\mathbf{R}\hat{i}+\mathbf{S}\hat{j}+\mathbf{T}\hat{k}\right)\cdot\mathbf{dS} \quad (1)$$

where, **q** is the flow variable vector,

$$\mathbf{q}=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{Bmatrix},$$

F, **G** and **H** are the inviscid fluxes,

$$\mathbf{F}=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho vu\\ \rho wu\\ \rho Eu+pu\end{Bmatrix},\mathbf{G}=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho wv\\ \rho Ev+pv\end{Bmatrix},\mathbf{H}=\begin{Bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ \rho Ew+pw\end{Bmatrix},$$

and **R**, **S** and **T** are the viscous fluxes,

$$\mathbf{R}=\begin{Bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ u\tau_{xx}+v\tau_{xy}\\ +w\tau_{xz}+kT_x\end{Bmatrix},\mathbf{S}=\begin{Bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{yz}\\ u\tau_{xy}+v\tau_{yy}\\ +w\tau_{yz}+kT_y\end{Bmatrix},\mathbf{T}=\begin{Bmatrix}0\\ \tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\\ u\tau_{xz}+v\tau_{yz}\\ +w\tau_{zz}+kT_z\end{Bmatrix}$$

In the above equations, ρ
w are the fluid velocity components; E is the total energy per volume; T is temperature; k is the
τ_{xx} τ_{xy} τ_{xz} τ_{yy}
τ_{yz} τ_{zz} are the viscous stress tensor components. For
modified to include the grid velocities. In this case, equation (1) would become

$$\frac{\partial}{\partial t}\iiint_{Vol}\mathbf{q}\,dVol+\oiint_{S}\left(\mathbf{F}\hat{i}+\mathbf{G}j\quad\mathbf{H}\hat{k}\right)\cdot\mathbf{dS}-\oiint_{S}\mathbf{q}\mathbf{V}_{grid}\ \mathbf{dS}=\oiint\left(\mathbf{R}\hat{i}+\mathbf{S}j\quad\mathbf{T}\hat{k}\right)\cdot\mathbf{dS}$$

The first integral of equation (1) is over the
q at the cell

are the summation over all the faces of the current cell of the

In semi-discrete form of equation (1) may be written as:

$$\frac{d(Vol\,\mathbf{q})}{}+\sum_{All\ faces}\left(\mathbf{F}i+\quad j+\mathbf{H}\ \right)\cdot\mathbf{dS}=\sum_{All\ faces}\left(\quad i+\mathbf{S}j+\mathbf{T}\ \right)\cdot$$

Vol term can be taken

this will be assumed to be the case here. However, the method has been developed for general deforming

meshes. In the present implicit scheme, the viscous fluxes are treated explicitly. Define,

$$\hat{\mathbf{F}} = \mathbf{F}\hat{i} + \mathbf{G}\hat{j} + \mathbf{H}\hat{k}$$

so that the semi-discrete Euler equations can be written as:

$$Vol\frac{d\mathbf{q}}{dt} + \sum_{All\ faces} \hat{\mathbf{F}}^{n+1}\cdot\hat{\mathbf{n}}\Delta S = \mathbf{R}_1 \tag{4}$$

where ΔS is the face area, $\mathbf{R}_1$ are the viscous contributions evaluated at the previous time level, and $\hat{\mathbf{n}}$ is the unit normal to that face.

In an implicit scheme, $\hat{\mathbf{F}}$ will be computed at the new time level *n+1*. Expanding $\hat{\mathbf{F}}$ about a known time level, *n*, we get:

$$\begin{aligned}\hat{\mathbf{F}}^{n+1} &= \hat{\mathbf{F}}^{n} + \left(\frac{\P\hat{\mathbf{F}}}{\P\mathbf{q}}\right)^{n}\left(\mathbf{q}^{n+1}-\mathbf{q}^{n}\right) + O\left(\Delta\mathbf{q}^2\right)\\ &= \hat{\mathbf{F}}^{n} + \hat{\mathbf{A}}\Delta\mathbf{q} + O\left(\Delta\mathbf{q}^2\right)\end{aligned} \tag{5}$$

where $\hat{\mathbf{A}} = \frac{d\hat{\mathbf{F}}}{d\mathbf{q}}$. Substituting equation (5) into (4), the semi-discrete form becomes:

$$Vol\frac{d\mathbf{q}}{dt} + \sum_{All\ faces}\hat{\mathbf{A}}\Delta\mathbf{q} = -\sum_{All\ faces}\hat{\mathbf{F}}^{n}\cdot\hat{\mathbf{n}}\Delta S + \mathbf{R}_1 \tag{6}$$

where the right hand side contains information only at the known time level and may be evaluated at the start of a given time step. In the present work, this was done using the existing Riemann solver in COBALT. This ensured the implicit solver and the explicit solver will yield identical steady state results.

The matrix $\hat{\mathbf{A}}$ has real eigenvalues and may be split into two matrices $\hat{\mathbf{A}}^{+}$ and $\hat{\mathbf{A}}^{-}$. In the present work, the matrix $\hat{\mathbf{A}}$ is split according to Jameson and Turkel[16], and Yoon and Kwak[17]:

$$\hat{\mathbf{A}}^{+} = \frac{\hat{\mathbf{A}} + \mathbf{I}b\,l_{\max}}{2}$$

$$\hat{\mathbf{A}}^{-} = \frac{\hat{\mathbf{A}} - \mathbf{I}b\,l_{\max}}{2}$$

where *b* is a relaxation factor and $l_{\max} = (\mathbf{V}\cdot\hat{\mathbf{n}} + c)$. Equation (6) now becomes:

$$Vol\frac{d\mathbf{q}}{dt} + \sum_{All\ faces}\left[\hat{\mathbf{A}}^{+}\Delta\mathbf{q}_{nc} + \hat{\mathbf{A}}^{-}\Delta\mathbf{q}_{neighbor}\right] = -\sum_{All\ faces}\hat{\mathbf{F}}^{n}\cdot\hat{\mathbf{n}}\Delta S + \mathbf{R}_1 \tag{7}$$

where *nc* is the current cell.

Note that the terms involving the factor *b* add a numerical viscosity, similar in spirit to the implicit viscosity present in alternating direction implicit schemes. The formal temporal accuracy of the scheme is not compromised, so long as *b* is of order unity. Furthermore, in iterative time marching schemes to be discussed later, the error introduced by this factor will be driven to zero as $\Delta\mathbf{q}$ is driven to zero.

Replacing the time derivative with a finite difference approximation, the semi-discrete form becomes:

$$\begin{aligned}Vol\frac{\Delta\mathbf{q}_{nc}}{\Delta t} + \sum_{All\ faces}\left[\hat{\mathbf{A}}^{+}\Delta\mathbf{q}_{nc} + \hat{\mathbf{A}}^{-}\Delta\mathbf{q}_{neighbor}\right] &= \mathbf{R}_1\\ &\quad - \sum_{All\ faces}\hat{\mathbf{F}}^{n}\cdot\hat{\mathbf{n}}\Delta S\end{aligned} \tag{8}$$

Note that the $\hat{\mathbf{A}}^{-}\Delta\mathbf{q}$ quantity represents the flux change associated with waves that have a negative wave speed, i.e. waves that enter the cell *nc* from outside. Only the *neighbor* cells can participate in this computation of the flux changes. Likewise, $\hat{\mathbf{A}}^{+}\Delta\mathbf{q}$ represents flux changes associated with waves of positive wave speed, i.e. waves which leave the cell *nc*. These flux changes should be computed using information only within the cell *nc*.

Since the present scheme is a point implicit scheme, the neighbor values are either know or set to zero. Therefore, the equation (8) becomes:

$$\begin{aligned}Vol\frac{\Delta\mathbf{q}_{nc}}{\Delta t} + \sum_{All\ faces}\hat{\mathbf{A}}^{+}\Delta\mathbf{q}_{nc} &= \mathbf{R}_1 - \sum_{All\ faces}\hat{\mathbf{F}}^{n}\cdot\hat{\mathbf{n}}\Delta S\\ &\quad - \sum_{All\ faces}\hat{\mathbf{A}}^{-}\Delta\mathbf{q}_{neighbor}\end{aligned} \tag{9}$$

or,

$$[\mathbf{M}]\Delta\mathbf{q}_{nc} = \{\mathbf{RHS}\}^{n} \tag{10}$$

This is a 5×5 matrix and diagonally dominant. It is easily inverted. the solution is updated by:

$$\mathbf{q}^{n,predictor} = \mathbf{q}^{n} + \Delta\mathbf{q} \tag{11}$$

During the second sweep, the elements are updated in the reverse order.

In order to improve convergence, the cells are locally reordered. In the present scheme, we sweep in a reorder direction to maximize the number of neighboring cells whose $\Delta\mathbf{q}$'s have been calculated before the current cell. This is achieved by starting at

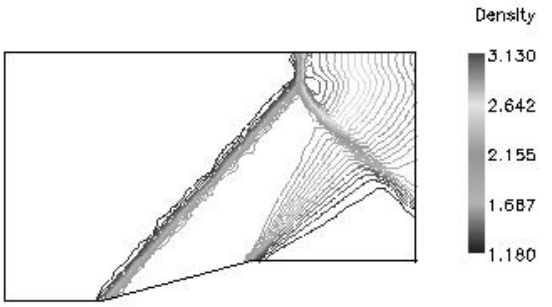

Fig 1a: Explicit Solution Density Contours for the Supersonic Ramp, M = 1.9

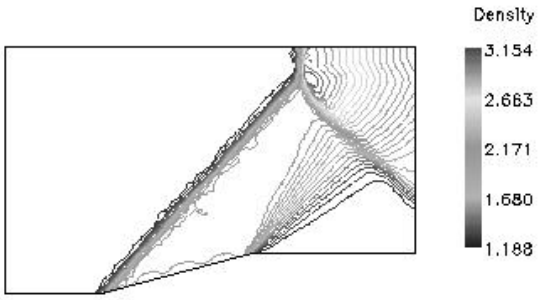

Fig 1b: Implicit Solution Density Contours for the Supersonic Ramp, M = 1.9

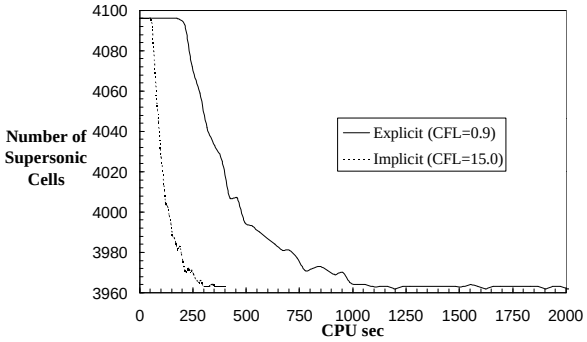

Fig 1c: Comparison of the Convergence of the Number of Supersonic Cells for the Supersonic Ramp, M = 1.9

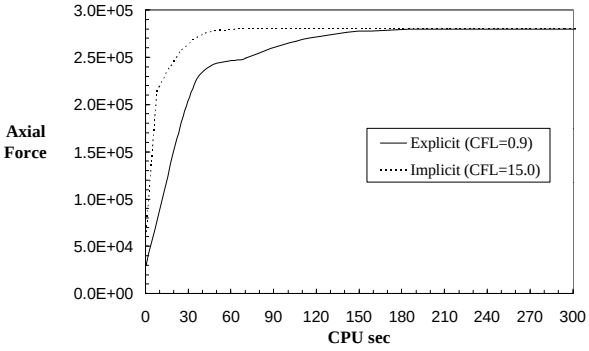

Fig 1d: Comparison of the Convergence of the Axial Force for the Supersonic Ramp, M = 1.9

a user specified cell. This cell number is added to a sweep direction list. All its neighbors are then added to the sweep list. Each entry in the sweep list has its neighbors checked; if a neighbor cell is not already in the sweep list, add it. This process is continued until all the cells are added to the sweep list. The first sweep follows the order of the sweep list; the second sweep follows the reverse order.

Iterative Time Marching Scheme:

For unsteady flows involving very large time steps, a slight variant to equation (6) is solved:

$$Vol\frac{\Delta\mathbf{q}^{m+1}}{\Delta t}+\sum_{All\ faces}\hat{\mathbf{A}}\Delta\mathbf{q}^{m+1}=\mathbf{R}_1-\sum_{All\ faces}\hat{\mathbf{F}}^m\cdot\hat{\mathbf{n}}\Delta S-Vol\frac{\mathbf{q}^m-\mathbf{q}^n}{\Delta t}\quad(12)$$

where $\Delta\mathbf{q}^{m+1}$ is viewed as changes in flow properties between successive iterations, *m*, rather than successive time steps *n*. The right hand side is the discrete approximation to the unsteady governing equations. At convergence of a iteration,

$$\mathbf{q}^{m+1}=\mathbf{q}^m=\mathbf{q}^{n+1}$$

and the discrete form is exactly satisfied. Errors such as linearization errors, the errors associated with the factor *b* and the LU approximate factorization errors all vanish.

Grid Deformation:

In the present study, for unsteady flows in which the grid is allowed to deform, the volume term is not brought outside of the time derivative. The corresponding discrete equation is:

$$\frac{\Delta(Vol\mathbf{q})^{m+1}}{\Delta t}+\sum_{All\ faces}\left(\frac{\hat{\mathbf{A}}}{Vol}\right)\Delta(Vol\mathbf{q})^{m+1}=-\sum_{All\ faces}\hat{\mathbf{F}}^m\cdot\hat{\mathbf{n}}\Delta S-\frac{(Vol\mathbf{q})^m-(Vol\mathbf{q})^n}{\Delta t}+\mathbf{R}_1\quad(13)$$

Once the $\Delta(Vol\mathbf{q})^{m+1}$ calculated, the flow variables are calculated by:

$$\mathbf{q}^{m+1}=\frac{\left[(Vol\mathbf{q})^m+\Delta(Vol\mathbf{q})^{m+1}\right]}{Vol^{m+1}}\quad(14)$$

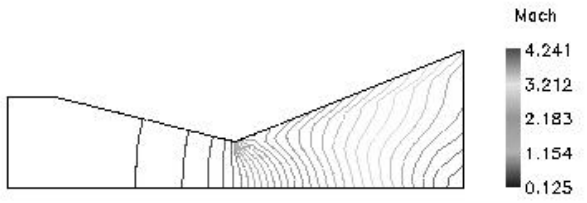

Fig 2a: Explicit Solution Mach Contours for the CD Nozzle, M = 0.1467

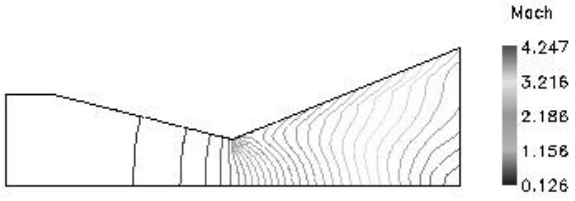

Fig 2b: Implicit Solution Mach Contours for the CD Nozzle, M = 0.1467

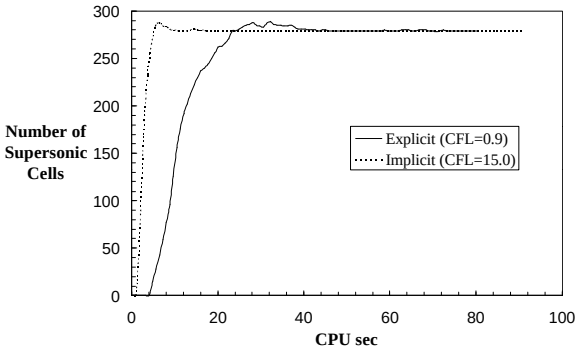

Fig 2c: Comparison of the Convergence of the Number of Supersonic Cells for the CD Nozzle, M = 0.1467

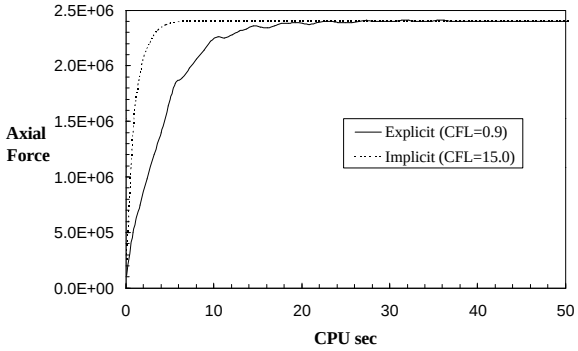

Fig 2d: Comparison of the Convergence of the Axial Force for the CD Nozzle, M = 0.1467

Results and Discussion

A wide range of tests cases has been run to validate the implicit version of COBALT and to show the improvement in computational efficiency gained over the original explicit algorithm. These test cases include both internal and external flows. Results are obtained for 2-D, 3-D, and axi-symmetric flows. Steady and unsteady simulations have been run and are compared with the explicit algorithm's solution or with experimental data.

Supersonic Ramp

The ramped channel is a simple supersonic inlet modeled with one straight wall and one ramped wall, with the inlet M = 1.9. The ramp angle was 14.93°. The grid contains 4,096 triangular cells The resulting shock has an oblique section and a normal section. The explicit solution is shown in figure 1a. The implicit solution is shown in figure 1b for comparison. It is clearly seen that the implicit algorithm converges to the same solution as the original explicit algorithm. Both solutions calculate the number of supersonic cells to be 3963 and the axial force to be 2.8×10^5 N as seen in figures 1c and 1d. The implicit algorithm converges in 1 minute of CPU time as compared to 3.2 minutes for the explicit algorithm. All runs were done on an IBM RS6000 workstation. No machine specific coding was done; therefore, a greater speed up could be accomplished if the machines architecture was taken into account. Notice that this problem has large subsonic and supersonic regions. A sweep against supersonic flow can be destabilizing; but, the opposite sweep stabilizes it. No unexpected difficulties were encountered.

Converging-Diverging Nozzle

This test case deals with choked flow through a converging-diverging axi-symmetric nozzle. Cobalt needs only one plane, x-r, of data to calculate axi-symmetric flow. The axi-symmetric form has a P/r term which has been added both to the right hand side of the explicit code and to the LU factorization in the radial direction momentum equation. The grid contains 784 quadrilateral cells. The inlet Mach number is M=0.1467. Figures 2a and 2b show Mach number contours for the explicit solution and the implicit solution, respectively. Clearly, both algorithms converge to the same solution. Both algorithms converge to 279 supersonic cells and an axial force of 2.4×10^6 N as shown in figures 2c and 2d.

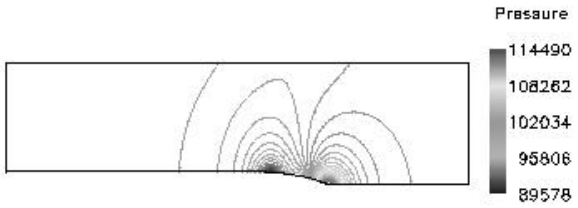

Fig 3a: Explicit Solution Pressure Contours for the AGARD Boat-Tail Nozzle, M = 0.8, Re = 18.64 Million

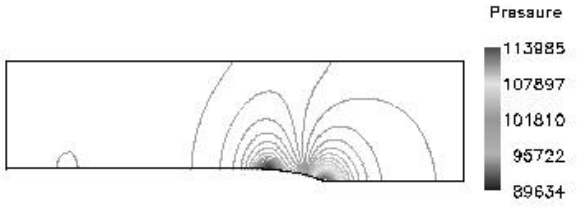

Fig 3b: Implicit Solution Pressure Contours for the AGARD Boat-Tail Nozzle M = 0.8, Re = 18.64 Million

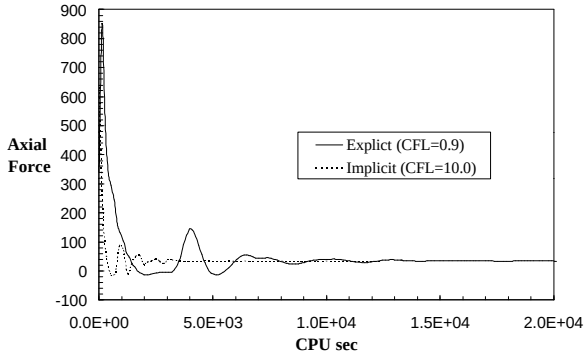

Fig 3c: Comparison of Convergence of Axial Force for the AGARD Boat-Tail Nozzle M = 0.8, Re = 18.64 Million

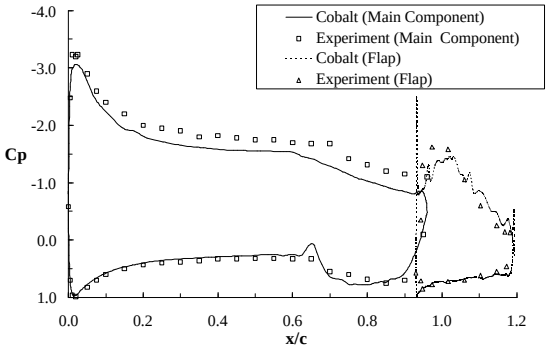

Fig 4a: Cp Distribution for the GA(W) -1 Airfoil for $\alpha = 0°$, $M_\infty = 0.14$

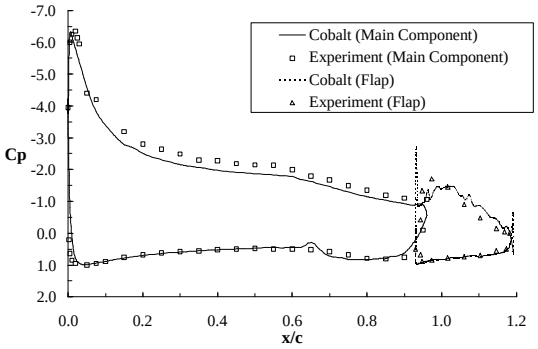

Fig 4b: Cp Distribution for the GA(W)-1 Airfoil for $\alpha = 5°$, $M_\infty = 0.14$

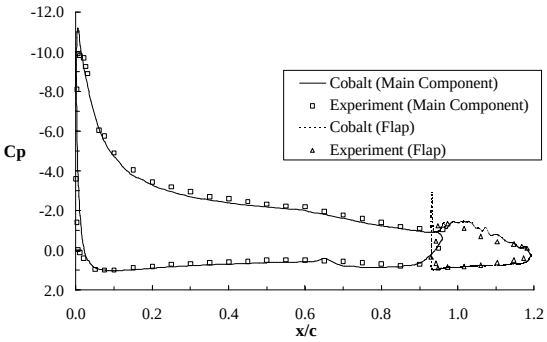

Fig 4c: Cp Distribution for the GA(W) -1 Airfoil for $\alpha = 10°$, $M_\infty = 0.14$

The implicit solution is converged in 11 CPU seconds, while the explicit solution requires 47 CPU seconds.

AGARD Boat-Tail Nozzle

In this test case, turbulent flow through an AGARD Boat-Tail axi-symmetric nozzle is simulated. The grid contains 13,452 quadrilateral cells. The inlet Mach number is M = 0.8, and the Reynolds number is 18.64 million. Implicit modeling of the turbulence effects is achieved by augmenting the diagonal term. In this case, the Spalart-Allmaras turbulence model is employed, and was integrated implicitly in time using a variant of the present LU scheme. Figures 3a and 3b show pressure contours for the explicit and implicit solution, respectively. Both solutions converge to the same answers. Figure 3c shows the convergence times for the explicit and implicit algorithms. Both solutions converge to an axial force of 33.5 N. However, the implicit algorithm does so in less than 1 hr of CPU time versus 3.5 hr of CPU time required by the explicit algorithm.

GA(W) -1 Airfoil

This airfoil was tested by Wentz and Seetharam[18] and is widely used for code validation. The present simulation uses the 29% chord fowler flap

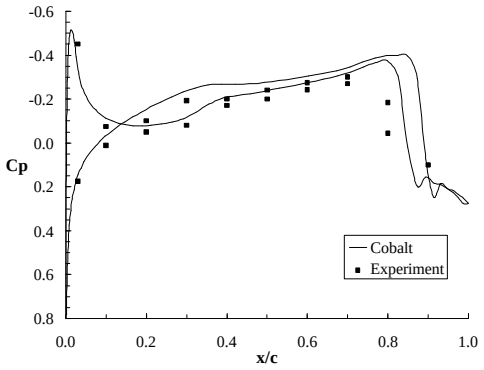

Fig 5a: Cp Distribution for the F5 Wing at 35.2% of the span, M_∞ = 0.95

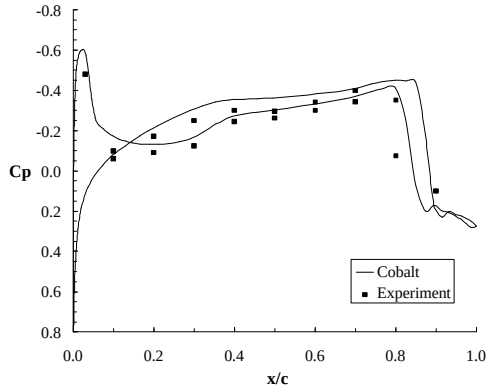

Fig 5b: Cp Distribution for the F5 Wing at 72.1% of the span, M_∞ = 0.95

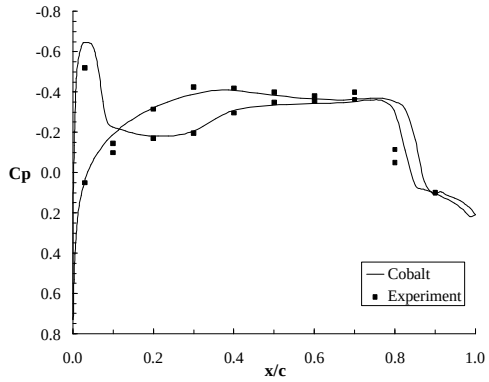

Fig 5c: Cp Distribution for the F5 Wing at 97.7% of the span, M_∞ = 0.95

set at 30° flap deflection angle. The grid contains 14,609 triangular cells. The simulation M_∞= 0.14 matches the experimental conditions. The experiment was run at Re = 2.2 million. For the present study, three angle of attacks, α=0°, α=5°, and α=10° were considered. The surface Cp distributions are plotted in figure 4a-c, respectively. Good correlation over the airfoil surface is observed. The suction peak magnitude is seen to be predicted, but the location is slightly forward of the experimental data. The slight bump in the Cp distribution on the lower surface at 0.6-0.7c is due to lack of geometric surface definition from the experiment in this region. Good correlation between the calculations and the experiments observed over the flap also.

F5 Wing

The F5 wing was tested by Tijdemann[19-20] at steady angles of attack and with a harmonic oscillatory motion for a wide range of Mach numbers. This case has been studied by a number of researchers, and makes an excellent test case due to the complexity of the geometry, and the combined bending-torsional deformations that must be simulated.

F5 Wing - Steady Flow Simulations

For the validation of the implicit algorithm in 3-D, steady transonic flow, the test case at M=0.95 was simulated. The grid was composed of 184,800 hexahedral cells. The angle of attack for this case was α = 0°. Figures 5a, b and c show the Cp distributions at spanwise location of 35.2%, 72.1% and 97.7% of the span. Good correlation is obtained at each span location. As may be expected, the present inviscid simulation predicts a shock that is slightly aft and slightly strong compared to the experiments.

F5 Wing - Unsteady Simulations

The F5 wing unsteady test case simulated is a harmonically oscillating surface. The movement of the wing surface is obtained using the mode shape found in Tidjemann[20]. This mode shape can be considered to be a rigid rotation of each spanwise section about a node line, as discussed by Weed[21] and Mello[22]. This simulation had a Mach number of M_∞= 0.95, a reduced frequency of k = 0.528 corresponding to the frequency of oscillation of 40 Hz, and an amplitude of oscillation of $\Delta\alpha$= 0.222°. The results of the first harmonic, real and imaginary components of the unsteady surface pressure coefficients were calculated using a Fourier series analysis. The real and imaginary Cp distributions, normalized by the amplitude of oscillation, at the 35.2%, 72.1% and 97.7% span stations are shown in figure 6a-f, respectively. This simulation was done with 240 time steps per cycle. The third cycle results are shown, which allows sufficient time for the transients to be convected out of the flow domain. The results compare well with experiments. The leading suction peak is sufficiently

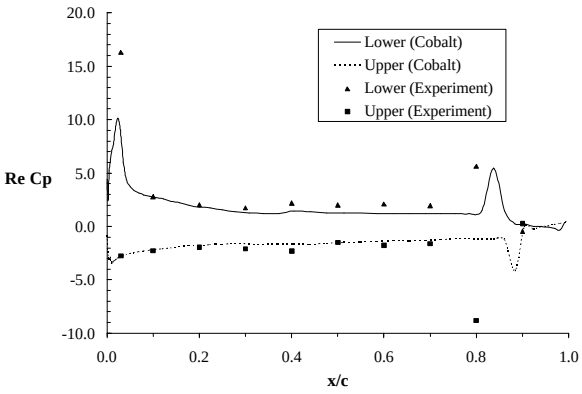

Fig 6a: Unsteady Pressure, Real Part at 35.2% span M_∞ = 0.95 and k = 0.528

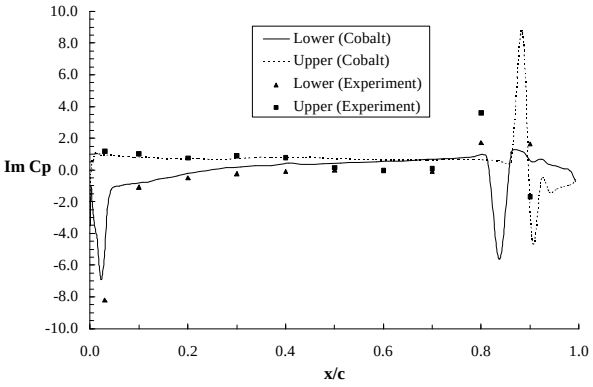

Fig 6b: Unsteady Pressure, Imaginary Part at 35.2% span M_∞ = 0.95 and k = 0.528

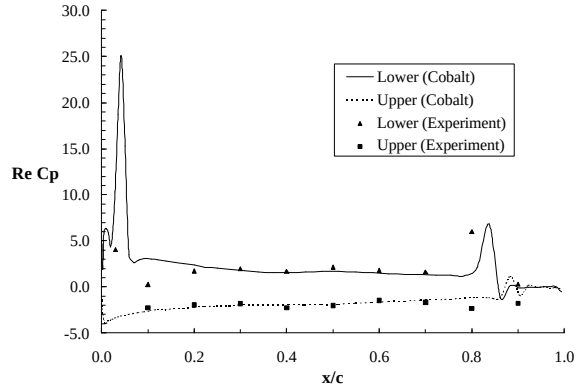

Fig 6c: Unsteady Pressure, Real Part at 72.1% span M_∞ = 0.95 and k = 0.528

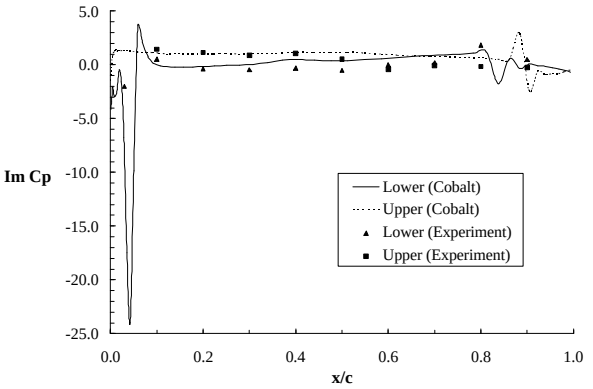

Fig 6d: Unsteady Pressure, Imaginary Part at 72.1% span M_∞ = 0.95 and k = 0.528

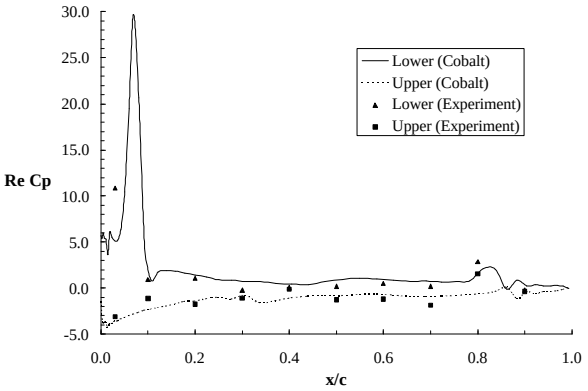

Fig 6e: Unsteady Pressure, Real Part at 97.7% span M_∞ = 0.95 and k = 0.528

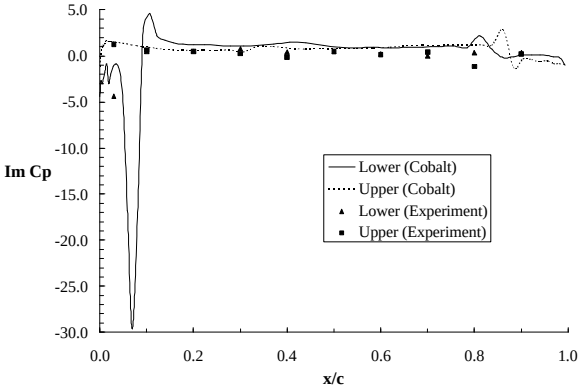

Fig 6f: Unsteady Pressure, Imaginary Part at 97.7% span M_∞ = 0.95 and k = 0.528

captured. The shock and its movement, shown as the bulge at around 0.8 x/c, is aft of the actual shock. This is to be expected for an Euler solution. However, the shock strength is too weak for an Euler solution.

Concluding Remarks

An implicit, unstructured grid based flow solver has been described, and validated for a number of test cases. Good agreement with measured data have been obtained for all the cases that have been tested. In many instances a factor of 3 or more speed up over explicit scheme was achieved.

Additional improvements to the flow solver are needed and are planned. Distributed computing can be achieved with a minimal change to the flow solver coding. A formal accuracy in space from 2nd to higher order using a multi-dimensional interpolations

can further enhance the code's ability to capture shocks and shear layers on sparse grids. To date, only two temporal validations have been done. Additional validations, particularly in the transonic and supersonic regimes are needed.

References

1. Batina, J. T., "Implicit Upwind Solution Algorithms for Three Dimensional Unstructured Meshes," *AIAA Journal*, Vol 31, no 5, May 1993, p. 801-805.

2. Billey, V., Periaux, J., Stoufflet, B., Derieux, A., Fezoui, L. and Selmin, V., "Recent Improvements in Galerkin and Upwind Euler Solvers and Application to Three Dimensional Transonic Flow in Aircraft Design," *Computer Methods in Applied Mechanics and Engineering*, Vol 75, October 1989, p 409-414.

3. Barth, T. J. and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0336, January 1989.

4. Newman, J. C., "Flow Simulations about Steady-Complex and Unsteady Moving Configurations Using Structured-Overlapped and Unstructured Grids," NASA-CR-199781, January 1995.

5. Thareja, R. R., Stewart, J. R., Hassan, O., Morgan, K. and Peraire, J., "A Point Implicit Unstructured Grid Solver For the Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol 9, 1989, p.405-425.

6. Hassan, O., Morgan, K. and Peraire, J., "An Implicit/Explicit Scheme for Compressible Viscous High Speed Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol 76, November 1989, p. 245-258.

7. Slack, D. C., Whitaker, D. L. and Walters, R. W., "Time Integration Algorithms for Two Dimensional Euler Equations on Unstructured Meshes," *AIAA Journal*, Vol 32, no 6, June 1994, p. 1158-1166.

8. Knight, D. D., "A Fully Implicit Navier-Stokes Algorithm using an Unstructured Grid and Flux Difference Splitting," *Applied Numerical Mathematics*, Vol 16, 1994, p 101-128.

9. Rausch, R. D., Batina, J. T. and Yang, H. T. Y., "Spatial Adaptation Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Computations," AIAA 91-1106, April 1991.

10. Venkatakrishnan, V. and Mavriplis, D. J., "Implicit Methods for the Computation of Unsteady Flows on Unstructured Grids," NASA-CR-198206, August 1995.

11. Venkatakrishnan, V. and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes," NASA-CR-187564, May 1991.

12. Collela, P., "Glimm's Method for Gas Dynamics," *SIAM J. Sci. Stat. Comput.*, Vol 3, No.1, 1982, p. 76-110.

13. Strang, W. Z., "Cobalt User's Manual," WL/FIMC WPAFB, OH 45433, February 1995.

14. Godunov, S. K., "A Finite Difference Method for the Numerical Calculation of Discontinuous Solutions of Equations of Fluid Dynamics," *Mat. Sb.*, Vol 47, 1959, p. 271-290.

15. Gottlieb, J. J. and Groth, C. P. T., "Assessment of Riemann Solvers for Unsteady One-Dimensional Inviscid Flows of Perfect Gases," *Journal of Computational Physics*, Vol 78, 1988, p. 437-458.

16. Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompostions," *Mathematics of Computation*, Vol 37, no 156, October 1981, p. 385-397.

17. Yoon, S. and Kwak, D., "Implicit Navier-Stokes Solver for Three Dimensional Compressible Flows," *AIAA Journal*, Vol 30, no 11, November 1992, p. 2653-2659.

18. Wentz, W. H., and Seetharam, H. C., "Development of Fowler Flap System for a High Performance General Aviation Airfoil," NASA-CR-2443, December 1974.

19. Tijdeman, H., van Nunen, J. W .G., Kraan, A. N., A. J., Poestkoke, R., Roos, R., Schippers, P., and Siebert, C. M., "Transonic Wind Tunnel Tests on an Oscillating Wing with External Stores, Part I: General Description," AFFDL-TR-78-194, Part I, December 1978.

20. Tijdeman, H., van Nunen, J. W .G., Kraan, A. N., A. J., Poestkoke, R., Roos, R., Schippers, P., and Siebert, C. M., "Transonic Wind Tunnel Tests on an Oscillating Wing with External Stores, Part II: Clean Wing," AFFDL-TR-78-194, Part II, March 1979.

21. Weed, R. A., “Computational Strategies for Three-Dimensional Flow Simulations on Distributed Computing Systems,” Ph D. Dissertation, School of Aerospace Engineering, Georgia Institute of Technology, August 1995.

22. Mello, O. A., “An Improved Hybrid Navier-Stokes/Full Potential Method for Computation of Unsteady Compressible Viscous Flows,” Ph D. Dissertation, School of Aerospace Engineering, Georgia Institute of Technology, November 1994.